Interspeech 2024
1-5 September 2024, Kos, Greece

Enhancing ECAPA-TDNN with Feature Processing Module and Attention Mechanism for Speaker Verification

Shiu-Hsiang Liou[1], *Po-Cheng Chan*[2], *Chia-Ping Chen*[1], *Tzu-Chieh Lin*[1], *Chung-Li Lu*[2], *Yu-Han Cheng*[2], *Hsiang-Feng Chuang*[2], *Wei-Yu Chen*[2]

[1]National Sun Yat-sen University, Taiwan [2]Advanced Technology Laboratory, Chunghwa Telecom Laboratories, Taipei, Taiwan

ddccbbaa1404@gmail.com, cbc@cht.com.tw, cpchen@cse.nsysu.edu.tw, 0201kawhi@gmail.com, {chungli, henacheng, gotop, weiweichen}@cht.com.tw

Abstract

In this paper, we introduce three methods to enhance the state-of-the-art ECAPA-TDNN model for speaker verification, namely self-calibration (SC), simple attention mechanism (SimAM), and a modified temporal dynamic convolution (MTDY) based front-end module. The SC module expands the model's receptive field and improves spatial attention for better capture of contextual information. The SimAM attention mechanism assigns unique weights to individual neurons, so it can place greater emphasis on more informative ones. The MDTY-based front-end module adapts itself to diverse temporal speech features with adaptive convolutional kernels, and aggregates these kernels to capture temporal variations with attention weights. Our proposed model, IM ECAPA MTDY-TDNN SimAM, demonstrates improved performance and complexity trade-offs compared to recent research works. On the VoxCeleb1-H test set, it achieves a 1.655% EER and 0.157 minDCF with 9.71M parameters and 1.97G FLOPs.

Index Terms: speaker verification, Time Delay Neural Network (TDNN), attention mechanism.

1. Introduction

Speaker verification (SV) is a task that identifies whether an input speech segment belongs to the claimed specific speaker. The method of authentication is usually based on biological cues in the acoustics, such as articulatory attributes and accents. In recent years, one of the advanced frameworks utilized in SV is the x-vector [1]. It utilizes Time Delay Neural Networks (TDNNs) [2] to extract features at the frame level and subsequently aggregate them into segment-level embedding. The current state-of-the-art (SOTA) in the field can be broadly categorized into two types of convolutional networks, namely the one-dimensional ECAPA-TDNN [3], and the two-dimensional ResNet [4]. The ECAPA-TDNN network incorporates the Res2Net [5] module and the squeeze-and-excitation (SE) [6] block. Res2Net is a multi-layer feature aggregation module utilized to integrate multi-scale features from shallow to deep. The SE block is responsible for assessing the significance of features on a channel-wise basis and generating corresponding weights. On the other hand, the ResNet network applies residual connections to enhance the robustness of its training. To improve the performance of the system, lots of research focuses on innovating modules such as ECAPA CNN-TDNN [7] and TDY-ResNet [8]. ECAPA CNN-TDNN is an enhanced version based on the ECAPA-TDNN backbone network. It enhances model performance by incorporating a 2D convolutional stem module to make tiny offsets in the frequency domain invariant. Kim et al. [8] showed that temporal and phonemic variation in speech is essential for identifying speaker characteristics. They proposed a temporal dynamic convolution (TDY) based on ResNet architecture to capture temporal variations information in speech. However, although the TDY architecture brings better model performance, it significantly increases the model complexity.

In this paper, we progressively enhance the ECAPA-TDNN architecture, with modifications divided into three parts. Firstly, we replace the Res2Net structure used in the ECAPA-TDNN with the self-calibration (SC) [9] module to enhance feature representation. It expands the model's receptive field with feature extraction in multiple spatial scales. Then we incorporate a CNN-based front-end module to preprocess the input audio. It employs 2D convolution with a small receptive field in both time and frequency dimensions to capture their relationship. Second, we adopt the SimAM attention mechanism [10] which assigns an individual weight to each neuron while considering frequency, time, and channel dimensions. Identifying the importance of each neuron is crucial for a neural network. SimAM attention mechanism allows us to focus on significant neurons, emphasizing important feature information. Finally, inspired by the previous study TDY [8], capturing phonemic variations in speech could enhance the robustness of speaker verification systems. However, due to the design of the TDY significantly increasing model complexity, we modify the structure of TDY to balance performance and computation trade-offs, naming it modified temporal dynamic convolution (MTDY). Then, we improve the convolutional structure in the CNN-based front-end module to the MTDY-based front-end module, utilizing its adaptive convolutional kernels to effectively capture temporal variations of phonemes in speech.

The rest of the paper is organized as follows: Section 2 introduces the baseline and improvement for the convolutional module and attention mechanisms. Section 3 focuses on the experiment setup including the dataset, training protocol, and evaluation protocol. Section 4 presents experimental results and associated discussion. Section 5 summarizes the contributions to the paper.

2. Network architecture

In this section, we introduce the baseline ECAPA-TDNN and provide a detailed description of the methods used to enhance it, including the improved version of the convolutional structure and attention mechanism.

2.1. Baseline model

ECAPA-TDNN [3] is a variant of Time Delay Neural Network (TDNN) that incorporates several advanced techniques such as SE-Res2Blocks, skip connections, and channel-dependent attentive statistics pooling. Owing to the multi-layer aggregation

10.21437/Interspeech.2024-601

strategy and multi-scale feature convolution, ECAPA-TDNN demonstrates outstanding performance. We re-implement ECAPA-TDNN as our baseline and gradually improve the model's performance through the steps described in the following subsections.

2.2. IM ECAPA CNN-TDNN

To enhance the feature extraction capability of the model, we refer to the previous study [11] and replace the Res2block used in ECAPA-TDNN with the self-calibration (SC) module. The improved model is denoted as IM ECAPA-TDNN, where "IM" is the abbreviation for improvement. Through the SC module, it can perform feature extraction in a heterogeneous manner and simultaneously integrate characteristics of different spatial scales. Diverging from Zhang et al. [11], we make adjustments to the architecture. The modifications include removing one layer of the aggregation structure and one layer of SE-SC-block. We utilize the output of the first TDNN structure as input for each subsequent aggregation layer. Our objective is to extract more speaker-specific information by combining the output vector from the first TDNN layer, which retains more original feature information.

Then, we incorporate a CNN-based front-end module to preprocess the input audio, naming the improved model IM ECAPA CNN-TDNN. The same speaker may exhibit slight frequency axis deviations due to different recording devices and environmental conditions. Consequently, the system may easily misclassify them as different speakers. To solve frequency axis deviation, we utilize 2D convolution with a small receptive field to extract local frequency and temporal information. It pre-processes the input audio to approximate the actual features of the speaker. Besides, Deng et al. [12] proposed that certain frequency features contain more information about speaker differentiation. Therefore, we incorporate an attention block after each Conv2D ResBlock in order to place greater emphasis on certain frequency features.

2.3. Simple attention mechanism (SimAM)

In visual neuroscience, neurons that exhibit distinct firing patterns from their surrounding neurons are typically considered the most informative. In order to identify more discriminative neurons, SimAM [10] was proposed to define an energy equation based on neuroscience discoveries. Neurons with lower energy function values are more discriminative and hold greater significance. Specifically, the energy function of neurons is defined by

$$e_t\left(w_t, b_t, y, x_i\right) = \left(y_t - \hat{t}\right)^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}\left(y_o - \hat{x_i}\right)^2 \quad (1)$$

$\hat{t} = w_t t + b_t$ and $\hat{x}_i = w_t x_i + b_t$ are linear transforms of the target neuron $\hat{t}$ and other neurons $\hat{x}_i$ in a single channel of input feature $X \in \mathbb{R}^{C\times F\times T}$. i is the index over the frequency-time domain and $M = F \times T$ represents the number of neurons on that channel. F and T represent the frequency and time dimensions, respectively. w_t and b_t are the weights and biases of linear transforms. y_t and y_o represent the values of the target neuron and other neurons, respectively. While $\hat{t}$ equals y_t and $\hat{x_i}$ equals y_o, the energy function meets its minimum value.

To simplify Eq. (1), we adopt a binary label setting where $y_t = 1$ and $y_o = -1$. Additionally, we add a regularizer, denoted as $\lambda\omega_t^2$, to prevent overfitting. The simplified energy function is represented by

$$\begin{aligned} e_t\left(w_t, b_t, y, x_i\right) = &\frac{1}{M-1}\sum_{i=1}^{M-1}\left(-1-\left(\omega_t x_i + b_t\right)\right)^2 \\ &+ \left(1-\left(w_t t + b_t\right)\right)^2 + \lambda w_t^2 \end{aligned} \quad (2)$$

Nevertheless, optimizing Eq. (2) through Adam or SGD involves a significant amount of computation. Fortunately, by taking derivatives of w_t and b_t, we obtain a closed-form solution to Eq. (3). The following represents simplified energy function

$$e_t^* = \frac{4\left(\hat{\sigma}^2 + \lambda\right)}{\left(t-\hat{u}\right)^2 + 2\hat{\sigma}^2 + 2\lambda} \quad (3)$$

$\hat{\mu}$ and $\hat{\sigma}^2$ are mean and variance over all neurons in that channel. So far, we derive an energy function where a smaller value of e_t^* corresponds to more importance for that neuron. The final energy function is represented by

$$\tilde{Y} = \sigma\left(\frac{1}{E}\right) \odot X \quad (4)$$

X is the input feature, and σ represents the sigmoid function. E assembles all energy values of e_t^* across channel, frequency, and time dimensions. Through the SimAM attention mechanism, the neural network is able to assign higher weights to more discriminative neurons, emphasizing important feature information. The network configuration of IM ECAPA CNN-TDNN SimAM is demonstrated in Fig. 1.

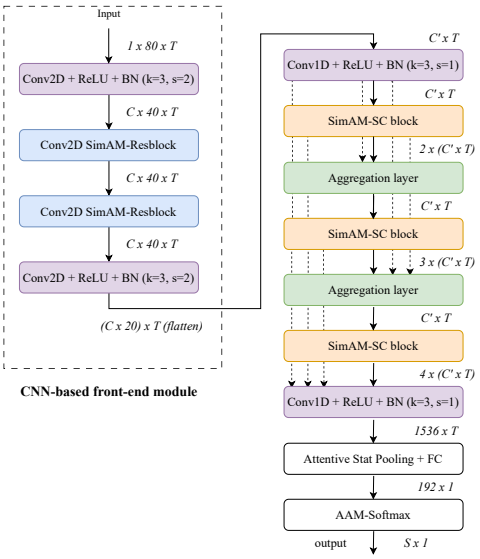

Figure 1: *The network architecture of IM ECAPA CNN-TDNN SimAM system. k represents the kernel size and s denotes the stride. C and C' correspond to the number of channels in the front-end module and the backbone structure, respectively. T represents the number of frames, and S indicates the number of speakers for classification.*

2.4. Modified temporal dynamic convolution (MTDY) based front-end module

It is intuitive to increase the number of convolutional layers in the network to improve feature extraction capability, but it significantly increases the system complexity. Chen et al. [13] introduced dynamic convolution (DY-CNN). It utilizes a set of

K parallel convolutional kernels which are dynamically aggregated through attention mechanisms. It improves the representation capability with a negligible increase in computation costs. Kim et al. [8] proposed a temporal dynamic convolutional neural network (TDY-CNN), which originates from the dynamic convolutional neural network (DY-CNN). It considers the temporal variation of speech by applying multiple kernels that adapt to each temporal segment. However, although their proposed architecture TDY-CNN enhances the model's performance, it substantially increases the number of required parameters. In order to strike a better balance between model performance and complexity, we propose a modified structure named MTDY which originates from the TDY architecture. The modification lies in the method of generating attention weights. The architecture of the MTDY module is illustrated in Fig. 2.

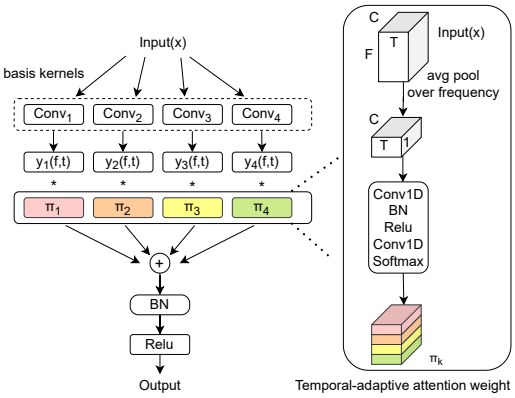

Figure 2: *An illustration of the modified temporal dynamic convolution (MTDY) module. C, F, and T represent the dimensions of the input feature for channel, frequency, and frame, respectively. π_k is the temporal-adaptive attention weight for k_{th} kernel.*

We follow the design of DY-CNN, where the input feature x is sent into a set of four parallel convolutional kernels. Subsequently, each output y_k from the basis kernel convolution is aggregated by attention weights π_k. π_k represents values that determine the importance of each basis kernel. Finally, we perform a weighted sum of the basis kernels with attention weights to automatically adjust the kernels, enabling them to capture the time-varying information in the utterance. Attention weight is built upon the input feature $x \in \mathbb{R}^{C\times F\times T}$. We use global average pooling over the frequency dimension of the input features to obtain the frequency descriptor for each frame. Then, two sets of 1D convolution are applied to take into account adjacent temporal components. The first 1D convolution layer reduces the channel dimension to one-fourth of the original. The second convolution layer further compresses the dimension of the channel to match the number of basis kernels. Finally, softmax is applied to constrain the values of temporal adaptive weights between 0 and 1 and ensure that the sum of attention weights for all basis kernels is equal to one.

We replace the convolutional structure in the CNN-based front-end module to the MTDY-based front-end module in order to improve the feature representation capability without significantly increasing the system complexity. The MTDY structure replaces the 2D convolution in the CNN-based front-end module. It utilizes adaptive convolutional kernels to adapt diverse temporal speech features and aggregates these kernels to effectively capture the temporal variations in speech. The configuration of the MTDY-based front-end module is depicted in Fig. 3.

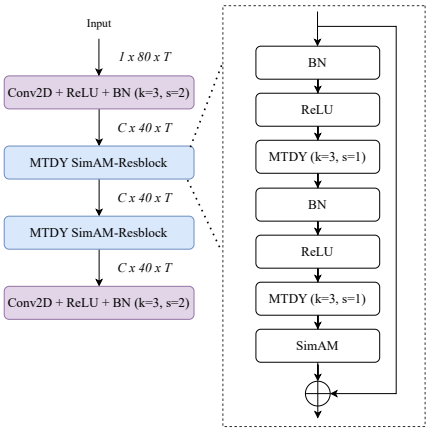

Figure 3: *The convolutional stem of the MTDY-based front-end module. C and T indicate the channel and frame dimensions of the input feature, respectively. k represents the kernel size and s denotes the stride.*

3. Experiment setup

In this section, we introduce the training and test datasets used in the experiment. We also provide a detailed description of various hyperparameter settings and explain the criteria to evaluate the model's performance.

3.1. Dataset and data augmentation

We utilize the development part of the VoxCeleb2 [15] dataset for training, which includes 5,994 speakers. The performance of models is evaluated on VoxCeleb1 [16]. We employ two types of data augmentation to increase the diversity of training data. One is the MUSAN dataset [17] used to add noise. The other is the RIR dataset [18] used to add reverberation. During each training stage, we randomly select one method from the original data without augmentation, the MUSAN corpus, or room impulse responses.

3.2. Training protocol

We employ 80-dimensional log Mel-filterbank energies (FBANK) from a 25 ms window and 10 ms frame shifts as input acoustic features. The features are cropped into 2-second segments, and each mini-batch comprises 256 segments. The Adam optimizer is used to adjust the neural network parameters, with an initial learning rate of 1e-3. The learning rate decays by 25% every 10 epochs. We adopt additive angular margin softmax (AAM-Softmax) [19, 20] as our loss function with a margin of 0.2 and a scale of 30. During training, weight decay is implemented with a value of 2e-5 to prevent overfitting. The hyperparameter λ is set to 1e-4. The channel size of the front-end module, denoted as C, is set to 64, while the channel size of the ECAPA-TDNN backbone, denoted as C', is set to 512. The embedding output layer is configured with 192 dimensions. Furthermore, we implement the Large Margin Fine-Tuning (LM-FT) method [21] to fine-tune all models. LM-FT parameter settings are as follows: margin of 0.5, frame size of 600, and learning rate of 3e-5.

Table 1: *Performance comparison in terms of EER (%), MinDCF, and model complexity among our proposed model and recent related works on the Voxceleb1 test set. The symbol of '-' in the table represents that the reference paper doesn't provide the information or the coefficient setting of evaluation metrics is different.*

Architecture	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H		Params	FLOPs
	EER(%)	MinDCF	EER(%)	MinDCF	EER(%)	MinDCF		
ECAPA-TDNN [3]	1.01	0.1274	1.24	0.1418	2.32	0.2181	6.2M	-
ResNet34 [3]	1.19	0.1592	1.33	0.1560	2.46	0.2288	23.9M	-
TDY-ResNet-34(×0.5) [8]	1.48	-	-	-	-	-	51.9M	-
MFA-TDNN [14]	0.856	0.092	1.083	0.118	2.049	0.190	7.32M	-
ECAPA-TDNN (re-implemented)	1.202	0.1128	1.279	0.1365	2.257	0.2095	**5.52M**	**0.93G**
IM ECAPA-TDNN	1.031	0.0988	1.218	0.131	2.21	0.2158	6.78M	1.19G
IM ECAPA CNN-TDNN	**0.798**	0.0963	1.082	0.1219	1.968	0.1835	10.04M	3.16G
IM ECAPA CNN-TDNN SimAM	0.915	0.092	0.968	0.0985	1.717	0.1627	9.84M	3.16G
Proposed: IM ECAPA MTDY-TDNN SimAM	**0.798**	**0.064**	**0.923**	**0.095**	**1.655**	**0.157**	9.71M	1.97G

3.3. Evaluation protocol

We evaluate the performance of our system by the equal error rate (EER) and the minimum detection cost function (MinDCF) with the settings of $P_{\text{target}} = 0.01$ and $C_{\text{fa}} = C_{\text{miss}} = 1$. The test set consists of the Voxceleb1-O/E/H sets. During the evaluation process, we calculate the cosine similarity score of the speaker embedding for each audio and apply adaptive score normalization (AS-Norm) [22, 23] for further normalization.

4. Results

In this section, we conduct a comparative analysis among the baseline and each improved version of the model. Table 1 presents the performance comparison among the baseline, the proposed architecture, and recent related works [3, 8, 14], considering aspects of both performance and model complexity. We follow the guidelines provided by Desplanques et al. [3] to re-implement the ECAPA-TDNN architecture as our baseline.

First of all, we replace the Res2Net structure in the baseline with the SC module. It splits convolutional filters into multiple portions and generates additional spatial attention at different scales that build a larger receptive field and increase abundant contextual information. IM ECAPA-TDNN improves model performance compared to the baseline, with a slight increase in model complexity. Moreover, we incorporate a CNN-based front-end module named IM ECAPA CNN-TDNN. It employs 2D convolution with a tiny receptive field to preprocess the input audio and effectively mitigates tiny frequency offsets, reducing noise generated by recordings in different environments. Although the incorporation of the CNN-based front-end module effectively enhances model performance, it significantly increases both parameter requirements and computational complexity. On the Voxceleb1-H test set, IM ECAPA CNN-TDNN outperforms the baseline with 12.8% and 12.4% relative improvement in EER and MinDCF.

Secondly, based on Yang et al. [10], they proposed an energy function to determine the importance of each neuron according to neuroscience theories. We replace the SE attention mechanism with the SimAM attention mechanism to make the neural network learn how to discriminate the importance of each neuron. Minimizing the energy function value of that neuron signifies its higher importance from the surrounding neurons, allowing us to place greater emphasis on the more informative ones. On the Voxceleb1-H test set, IM ECAPA CNN-TDNN SimAM outperforms IM ECAPA CNN-TDNN, showing a 12.8% and 11.3% relative improvement in EER and MinDCF, respectively.

Lastly, we replace the 2D convolution in the CNN-based front-end module with the MTDY module, which is based on the TDY architecture. According to Kim et al. [8], although replacing traditional convolution with TDY can enhance feature extraction capability, it significantly increases the model parameters. Therefore, we make adjustments to this architecture, specifically modifying the way of generating attention weights. The basis kernel weights of TDY are generated with two fully connected layers followed by input flattened along the channel and frequency dimensions. On the other hand, MTDY obtains the weights for each basis kernel by performing global average pooling over the frequency dimension of the input feature to obtain the frequency descriptor for each frame and then applies two sets of Conv1D. Thus, by modifying the way of generating attention weights, we not only benefit from the advantages of TDY but also significantly reduce the complexity of the model. Furthermore, compared to the CNN-based front-end module, MTDY adopts a set of four parallel convolutional kernels and aggregates these kernels with attention weights. The structure of the MTDY-based front-end module not only significantly reduces the model's complexity but also adapts itself to diverse temporal speech features to capture the temporal variations in speech. On the Voxceleb1-O test set, IM ECAPA MTDY-TDNN SimAM outperforms IM ECAPA CNN-TDNN SimAM, showing a relative improvement of 12.8% and 30.4% in EER and MinDCF, respectively, while also achieving a notable relative reduction in the required computation cost by 37.7%. Our proposed system, IM ECAPA MTDY-TDNN SimAM, not only demonstrates significant improvements over the baseline in system performance but also proves to be highly competitive compared to recent related works.

5. Conclusions

In this paper, we propose the IM ECAPA MTDY-TDNN SimAM model as an enhancement over the baseline, incorporating improved convolutional modules and attention mechanisms. The feature extraction capability is enhanced by utilizing the SC module and MTDY-based front-end module. The SimAM attention mechanism assigns unique weights to each neuron, emphasizing those with important feature information. Our proposed model, IM ECAPA MTDY-TDNN SimAM, demonstrates significant improvements compared to the baseline on the Voxceleb1-O test set, achieving a 33.6% relative improvement in EER and a 43.3% relative enhancement in MinDCF.

6. References

[1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[2] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth annual conference of the international speech communication association*, 2015.

[3] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp. 652–662, 2019.

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[7] J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating frequency translational invariance in tdnns and frequency positional information in 2d resnets to enhance speaker verification," *arXiv preprint arXiv:2104.02370*, 2021.

[8] S.-H. Kim, H. Nam, and Y.-H. Park, "Temporal dynamic convolutional neural network for text-independent speaker verification and phonemic analysis," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6742–6746.

[9] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 096–10 105.

[10] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, "Simam: A simple, parameter-free attention module for convolutional neural networks," in *International conference on machine learning*. PMLR, 2021, pp. 11 863–11 874.

[11] Y.-J. Zhang, Y.-W. Wang, C.-P. Chen, C.-L. Lu, and B.-C. Chan, "Improving time delay neural network based speaker recognition with convolutional block and feature aggregation methods." in *Interspeech*, 2021, pp. 76–80.

[12] A. Deng, S. Wang, W. Kang, and F. Deng, "On the importance of different frequency bins for speaker verification," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7537–7541.

[13] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 030–11 039.

[14] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Mfa: Tdnn with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7517–7521.

[15] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[16] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[17] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[18] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[19] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[20] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1652–1656.

[21] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5814–5818.

[22] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, M. D. Sánchez, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition." in *Interspeech*, 2017, pp. 1567–1571.

[23] S. Cumani, P. D. Batzu, D. Colibro, C. Vair, P. Laface, and V. Vasilakakis, "Comparison of speaker recognition approaches for real applications." in *INTERSPEECH*, 2011, pp. 2365–2368.